Enriching SMIL with Assertions for Temporal Validation

Annalisa Bossi  
Dept. of Computer Science  
University Ca’ Foscari of Venice  
via Torino, 155  
30172 Mestre (Venice) Italy  
bossi@dsi.unive.it

Ombretta Gaggi  
Dept. of Pure and Applied Mathematics  
University of Padua  
via Trieste, 63  
35121 Padua, Italy  
gaggi@math.unipd.it

ABSTRACT

In this paper we define a formal semantics for the language SMIL which can be used in a number of applications. First of all, we propose a computer-aided authoring system which include a Semantic Validator Module for the evaluation of the temporal consistency of the resulting multimedia presentation. If any temporal conflict is found, the system returns to the user a message pointing out the tag which contains the error and its motivation. This helps the user to correct the error. We also introduce a notion of equivalence for SMIL tags which is useful to find a candidate for substitution in the development of complex multimedia structure, for example in the context adaptation process.

Categories and Subject Descriptors

H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia—Theory; I.7.2 [Document and Text Processing]: Document Preparation—Standards, Markup Languages

General Terms

Theory, Verification

Keywords

SMIL, authoring, consistency checking

1. INTRODUCTION

A multimedia presentation is the best way to convey information to the user in many advanced applications like distance learning, virtual tourism, news delivery, entertainment and so on. A multimedia presentation is a collection of continuous media, like video or audio files, and static objects, like text pages and images, which can be distributed across the network and rendered to the user according to the author specifications. To be played, media objects must be disposed in the user screen and synchronized according to a time scale; therefore, the author of a presentation must define both the spatial layout and the temporal behavior of the document.

Multimedia authoring and design is a complex and error-prone activity, especially when the complexity of the temporal structure of multimedia documents increases together with the chance of including a temporal conflict in the synchronization constraints. Many researches address the problem of the specification of a multimedia presentation defining models ([1], [7], [14]), languages and tools ([2], [11], [18]). All the proposed solutions can be divided into two main classes (see [4]): the operational approach, which defines system-dependent structures to model the multimedia presentation, and the constraint-based approach. This second approach is more flexible but it requires the author to have in mind the overall structure of the final presentation which is not represented explicitly. This problem has found a partial solution with the structure of the tags proposed by the Synchronized Multimedia Integration Language, SMIL [9]. In fact, the most part of both the operational and the constraint-based systems use SMIL for building the final presentation.

Since SMIL’s first appearance, many authoring tools and players have been implemented, offering to their users different facilities like visual editors or preview windows, and sometimes, tools to check the correctness of the work-in-progress multimedia presentation. Unfortunately, most of the tools developed for SMIL check only the syntactic correctness of the document before playback, and are not able to find out semantic errors.

Motivating Scenario

A semantic error is a conflict in the temporal definition of the presentation: there are almost two conflicting values in the definition of the temporal attributes of the document. Usually, syntactic errors can be automatically corrected, whereas a semantic conflict points out a contradiction in the definition of the behavior of media items and requires a decision for its solution, which can be built-in in the system [16] or asked to the author. Unfortunately, as described in [10] and [17, 19], in presence of temporal conflicts, even simple multimedia documents may have different behaviors according to the chosen player, therefore the final behavior is almost unpredictable, as reported also by Eidenberger [5].

Let us consider two very common semantic errors: (1) wrong definition of attributes of the same object, e.g., the tag <text id="txt01" begin="3s" end="5s" dur="5s"/> defines a text message txt01 displayed at time instant 3 and removed after 2 time units, but the duration defined
by the attribute \texttt{dur} is equal to 5 seconds, and (2) wrong
definition of the temporal structure of the document, e. g.,
when the conflicting values involve more than one tag. As
an example, the tag
\begin{verbatim}
<seq dur="5s">
  <img id="img01" dur="5s" />
  <img id="img02" dur="5s" />
</seq>
\end{verbatim}
describes a sequence of two images, each one visualized for 5
seconds. The conflict is between the overall duration of the
sequence, i. e., 5 seconds according to its attribute \texttt{dur}, and
the sum of the durations of the single images it contains.

More common players, e. g. GRiNS [12] or RealPlayer [15],
(but not all) do not point out the temporal conflict to the
user but the playback goes on and the duration of an object
is equal to the minimum duration defined.

This means that, in the first case, text \texttt{txt01} lasts for
two seconds, and in the second case, image \texttt{img02} is not
displayed at all and the presentation ends immediately after
the first image. Possibly, this is exactly what the author
expects but, if not, it is important to individuate the existence
of the semantic conflict. Moreover, many players have prob-
lems in the resolution of the start and end time of media
items, especially when the complexity of the presentation
structure increases. Therefore, errors of the second kind are
very difficult to find out and fixed, when the complexity of
the presentation increases, because it is not always clear if
the misbehavior is due to a semantic conflict or to a bug in
the player [5].

Some authors, even identifying in the lack of a formal se-
manics one of the key problems of SMIL language (e. g.
Jourdan in [10]), do not consider the example above as tem-
poral conflicts, but define a formal semantics which describes
the behavior adopted by the majority of the existing play-
ers. Others, (e. g. Sampaio et al in [17]) use formal methods
to find out temporal conflicts like the ones described in this
paper. We agree with this second group of authors thus
considering a double inconsistent definition as a semantic
error since it points out a contradiction in the description of
the result the author of the multimedia presentation tries
to obtain. In the examples above, we are not sure that the
short interval time of 2 seconds is sufficient to read message
\texttt{txt01} and the author would not have included \texttt{img02} in the
presentation if her/his real intention is not to display it.

Summing up, consistency checking is an important issue
for multimedia documents and all their applications and any
authoring system should consider this aspect to guarantee
the generation of a renderable multimedia presentation. We
must note here that this paper does not aim at augmenting
or correcting the standard SMIL, but at offering a formal
semantics which can help guide SMIL developers, thus
improving the standard specification: as reported in [10] by
Muriel Jourdan, one of the editors of the SMIL 2.0 Timing
and Synchronization Module [13], “... SMIL 2.0 complexity
is so great that rejecting the use of formal supports gives rise
to a difficult-to-read specification that cannot be free from in-
consistency”.

Our proposal and its practical applications

In this paper we define a formal semantics for the language
SMIL 2.1 which is the basis for a \textit{Semantic Validator Mod-
ule} included into the authoring system depicted in Figure
1. Since an uncorrect multimedia presentation cannot be
rendered properly, consistency is checked during all the au-
thoring phase, each time the author asks it or when she/he
saves her/his work. We prefer this solution instead of a dy-
namic checking since we allow temporary inconsistencies due
to the \textit{work-in-progress} but we guarantee the correctness of
the final result. This choice is also cost-effective. The Se-
manic Validator Module supports the multimedia author-
ning process since, if any temporal inconsistency is found, it
returns to the user a message containing the tag which con-
tains the error and its motivation; e. g., the tool’s message for (1)
is "the text \texttt{txt01} contains two conflicting values for the
attributes end and dur", and for (2) "\texttt{img02} ends at time instant
10 but its father ends at time instant 5". These mes-
ages allow the user to easily detect and correct the errors.

Our semantics is defined by means of a set of inference
rules inspired by Hoare logic ([8]). The central feature of
Hoare logic is the Hoare triple which describes how the exe-
cution of a piece of code changes the state of the com-
putation. This choice brings the advantage that the SMIL
structure can be enriched by assertions, expressing the tem-
poral properties, which can be used during the authoring
phase, when media items are collected in more complex con-
structs. As an example, our tool can verify the consistency
of a multimedia presentation resulting from a context adap-
tation process. In this case, the document is dynamically
build up by selecting media items compatible with the great
number of different situations in which a multimedia presen-
tation can be played, in term of resources availability (e. g.,
network bandwidth, CPU time), device type (e. g., desktop,

laptop, cell-phone) and properties (e. g., screen size, number
of colors). This process may often generate conflicts which
must be solved in order to guarantee the playback.

Another application of our semantic approach derives from
the introduction of a formal notion of equivalence which
guarantees that two sets of tags can be substituted each
other without changing the behavior of the overall presen-
tation, or generating temporal conflicts. This notion can
be used in the context adaptation problem to choose the
candidate for a substitution.

As we compose a multimedia presentation by nesting a
SMIL tag into another, our rules allow us to compose the
semantics by evaluating a single tag inside a more complex
nesting. In other words, the proposed semantics is composi-
tional and helps the author to modularize her/his work thus
mastering the complexity of the verification of a multimedia
presentation consistency.

Despite SMIL 2 specification was first released in 2001,
most available players are often unstable or not free of charge
as reported in [5]. The major problem is a robust resolution
of start and end time of tags. Therefore, another application

Figure 1: The system architecture
of our Semantic Validator Module is as the basis for the development of an efficient player for SMIL documents, since it can detect presentations containing conflicts, thus avoiding to start their playback, moreover, if the document is consistent, it generates as output the correct begin and end time of every media item, which can be used for playback.

2. PRELIMINARIES

This paper presents a tentative approach to the formulation of a semantics for the verification of SMIL 2.1 tags using a formal system based on Hoare logic. In this section we start by introducing the basic elements and notations used through the paper.

We note here that our framework currently does not consider the whole SMIL language, even if the values for attributes which are missing is a very limited subset of the one allowed by the standard. Table 1 describes the set of tags, attributes and their possible values addressed in this paper. Note that we do not consider some user interaction like following a link to another document, or stopping the current playback. We plan to fill this gap in our future work.

2.1 The assertion language

In the next section we introduce a set of inference rules which describe how the execution of a piece of SMIL code changes the state of the playback. The rules provide an axiomatic semantics for the temporal aspects of SMIL tags in the spirit of Hoare logic. Therefore they allow us to derive judgements in the form of triplets:

\[ \{ P \} t \{ Q \} \]

where \( P \) and \( Q \) are assertions, respectively the precondition and the postcondition, and \( t \) is a SMIL tag. The triple \( \{ P \} t \{ Q \} \) can be read as: whenever the evaluation of the tag \( t \) starts in a state which satisfies the assertion \( P \) then it terminates in a state which satisfies the assertion \( Q \).

Since we are interested in describing only those aspects that might influence temporal consistency, a state describes only significant time instants: the start and end time instants of all SMIL tags contained in the presentation as well as the duration of each continuous object. Hence the assertion language used to express pre/post conditions includes a set of basic functions representing the significant temporal aspects of the media. Assertions are formed by sets of constraints on values returned by these functions. We say that \( P \) holds in a state \( \sigma \) or, equivalently, that \( \sigma \) satisfies \( P \) if all the constraints contained in \( P \) are true in \( \sigma \) and we write \( P \Rightarrow Q \) if \( Q \) holds in any state which satisfies \( P \).

<table>
<thead>
<tr>
<th>SMIL tags</th>
<th>text, img, video, audio, animation, brush par, seq, excl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>begin, end, dur</td>
</tr>
<tr>
<td>Admitted Values</td>
<td>begin: t, accesskey('c')+t, begin(m)+t, end(m)+t, end: t, accesskey('c')+t, begin(m)+t, end(m)+t, indefinite dur: t, indefinite</td>
</tr>
</tbody>
</table>

Table 1: List of SMIL tags and attributes used in this work

The only piece of useful information which is not contained in the SMIL document is the natural duration of each continuous media. We define the function \( t_{cr} : \text{Id} \rightarrow \mathbb{N} \) which takes as input an identifier of a continuous media \( (id \in \text{Id}) \) and returns the number of time instants for which the continuous media item plays in absence of user interaction or other temporal specification.

Our assertions also contain the function \( t_{cr} : \text{Id} \rightarrow \mathbb{N} \) that returns the current time instant in which the SMIL tag \( id \) is evaluated. By current time instant of a tag \( id \) we mean the time instant in which, considering a player executing the presentation, the player evaluates that command. As an example, if more than one media is defined inside a tag \( \text{par} \), they are evaluated at the same time instant, because they will be played in parallel (unless other attributes specification). Otherwise, in case of sequential composition, the player considers each media item one after the other, therefore the current time instant of each media is the end time of the previous one. This function is used to indicate which is the first tag of the document, i.e. the tag \( t \) with \( t_{cr}(t) = 0 \), or to evaluate a single tag if we want to check the correctness of only a portion of a presentation.

As a general remark, in the triple \( \{ P \} t \{ Q \} \) the precondition \( P \) contains, among others, the current time instant of the tag \( t \) and the natural duration of media items which it defines (if applicable). The postcondition \( Q \) contains, among

<table>
<thead>
<tr>
<th>Function</th>
<th>Where</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{cr} : \text{Id} \rightarrow \mathbb{N} )</td>
<td>Pre</td>
<td>returns the current time instant in which the SMIL tag ( id ) is evaluated</td>
</tr>
<tr>
<td>( \text{dur} : \text{Id} \rightarrow \mathbb{N} )</td>
<td>Pre</td>
<td>returns the number of time instants for which a continuous media item plays</td>
</tr>
<tr>
<td>( \text{begin} : \text{Id} \rightarrow \mathbb{N} )</td>
<td>Post</td>
<td>denotes the time instant media item ( id ) starts</td>
</tr>
<tr>
<td>( \text{end} : \text{Id} \rightarrow \mathbb{N} )</td>
<td>Post</td>
<td>denotes the time instant media item ( id ) ends</td>
</tr>
</tbody>
</table>

Table 2: List of functions and notations used in the definition of the proof rules
and sets whose description can be found in Table 4.

SMIL document with respect to the values of their attributes indicates to check the existence of an attribute's value before repetition of very similar rules, but we need a set of pred-

Table 3: List of abbreviations used in the definition of the proof rules

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>finite(k)</td>
<td>holds if k is a real value</td>
</tr>
<tr>
<td>indefinite(k)</td>
<td>holds if k is equal to ‘‘indefinite’’</td>
</tr>
<tr>
<td>defined(k)</td>
<td>holds if k is not equal to ‘‘void’’, i.e., finite(k) \lor indefinite(k) holds</td>
</tr>
<tr>
<td>NotDur</td>
<td>contains all the statements that have their attributes dur and end equal to void</td>
</tr>
<tr>
<td>Closure(c)</td>
<td>contains all the statements defined inside the tag c, at any level of nesting</td>
</tr>
<tr>
<td>Indef(c)</td>
<td>holds if in Closure(c) there are tags with attribute end (or dur) equal to ‘‘indefinite’’</td>
</tr>
</tbody>
</table>

Table 4: List of predicates and sets used in the definition of the proof rules

others, the definition of the time instants in which the tags defined in t begin and/or end. Media items definitions are evaluated through axioms, while for par and seq composition more complex rules are needed.

2.2 Notational conventions

In the following section we use a number of special notational conventions to introduce the set of inference rules describing the semantics of the SMIL tags.

Table 3 lists a set of abbreviations used for the representation of the SMIL tags. For instance <cmd c> stands for any tag SMIL with the attribute id = ‘‘c’’.

Moreover, we use the general form end=’’k1’’ and dur=’’k2’’ to represent the attributes of a tag where the meta-variable k is either any of the admitted values for the particular attribute, or the special value void. The value void represents the absence of that attribute and allows us to define only one rule for each compound tag. As regards the attribute begin we assume it is always defined since its absence can be represented by the value k=’’0’’.

For instance, <video id=’’v’’ begin=’’0’’ dur=’’5’’ end=’’void’’/> is considered as a synonymous of <video id=’’v’’ dur=’’5’’/>.

The advantage of this representation is that of avoiding repetition of very similar rules, but we need a set of predicates to check the existence of an attribute’s value before using it. We need also to classify the tags which occur in a SMIL document with respect to the values of their attributes dur and end. Hence we introduce some auxiliary predicates and sets whose description can be found in Table 4.

3. A SEMANTICS FOR SMIL TAGS

SMIL language definition provided by [9] does not contain a formal specification of tags and attributes semantics. The recommendation is divided into sections, some of which are defined “normative”. Sometimes, an algorithm is provided to better explain how significant time instants are computed, but neither a formal definition nor verification tools have been implemented by Synchronized Multimedia Working group of W3C to check the semantic correctness of SMIL tags.

In this section, we define a formal system which is able to find out temporal conflicts of a multimedia presentation defined using SMIL. The system provides a Hoare logic for SMIL by a set of inference rules describing how the execution of a piece of code changes the state of the playback.

We start by considering self contained tags, i. e., SMIL commands whose synchronization do not refer to other media items or tags. Axioms to verify the correctness of statements which define media items are listed in Table 5. The use of events is discussed in Section 3.2.

Assume we want to verify the triple:

\{P\}<video id=’’v’’ begin=’’2’’/>\{Q\}

where the precondition P is \{dur(v) = 5, t_{cr}(v) = 0\} and the postcondition Q is \{begin(v) = 2, end(v) = 7\}. The system verifies its correctness since, by applying the axiom CONT+BEGIN, it obtains the set of constraints \{t_{cr}(v) = 2, dur(v) = 7\} which is equivalent to the precondition.

The system can also be used as the basis for the implementation of a player. In this case, it applies the axiom

Table 5: Proof rules for media items definitions
CONT+BEGIN starting from the precondition, obtaining the postcondition that can be used to start and stop the video.

The situation is a little more complicated if the media definition contains also an end, or dur, attribute. The rule 

`MEDIA+BEGIN+END+dur` defines the end time of a media item if both \( k2 \) and \( k3 \) are finite (if defined), i.e., they are not equal to “\( \text{indefinite} \)”. As an example, we can apply the rule to verify that the triple

\[
\{P\} <\text{video id='v'} \text{begin='2'} \text{end='3'/>}\{Q\}
\]

where \( \{Q\} = \{\text{begin}(v) = 2, \text{end}(v) = 3\} \) is valid. As discussed in Section 2.2, `<video id='v'begin='2'end='3'/>` is a synonymous of `<video id='v'begin='2'end='3'/>`. For readability sake, we will use always this form in the following.

Media items definition does not lead to temporal conflicts unless the author defines both the dur and the end attributes. The applicability condition disallows the application of the rule in presence of incorrect values of these attributes; e. g., when both the attributes dur and end are finite, the relation \( k3 = k2 - k1 \) must hold. The applicability conditions also point out a temporal conflict to the user.

### 3.1 Rules for more complex constructs

When media definitions are nested into parallel and sequential composition, the evaluation of these structures requires the definition of more complex rules.

Since the flexibility of SMIL tags allows us to describe the same temporal behavior using both a par or a seq tag, we base the discussion of this section mainly on the description of the rules for the parallel composition. The sequential composition is discussed at the end of this section.

**PAR+BEGIN+END**

\[
\{A_i \cup \{t_{cr}(c_i) = \text{init}_c\} \} \subseteq \{B_i\} \quad \forall i \leq i \leq n
\]

\[
\{A'\} <\text{par c attribute-list}> c_1 \ldots c_n </\text{par}> \{B\}
\]

where

- \( \text{attribute-list} \equiv \text{begin='k1'} \text{end='k2'} \text{ dur='void'} \)
- \( A' = \bigcup_{i=1}^{n} A_i \cup \{t_{cr}(c) = \text{init}\} \)
- \( \text{init}_c = \text{init} + k1 \)
- \( B = \bigcup_{i=1}^{n} B_i \cup \{\text{begin}(c) = \text{init} + k1\} \cup \text{End} \)
- \( \text{stop} = \{\text{init} + k2 \}
\]

\[
\text{if finite}(k2) \\
\text{max}_c \{\text{end}_B(c_i)\} \text{ if } \neg \text{defined}(k2)
\]

- \( \text{End} = \{\text{end}(c) = \text{stop}\} \text{ if } \neg \text{Indef}(c) \\
\text{otherwise} \)

- \( B' = \bigcup_{i=1}^{n} B_i \)

**APPLICABILITY CONDITION:**

\[
\text{finite}(k2) \implies \neg \text{Indef}(c) \\
\text{Indef}(c) \implies (\neg \text{defined}(k2) \lor \text{indefinite}(k2)) \\
\text{finite}(k2) \implies \forall c_i \text{ end}_B(c_i) \leq \text{stop} \lor c_i \in \text{NotDur} \\
\land \forall c_i \text{ begin}_B(c_i) \geq \text{init} + k1
\]

Table 6: Proof rule for the parallel composition when the attribute \( \text{dur} \) is equal to \( \text{void} \)

We start our analysis by considering the parallel composition expressed by the tag par when the attribute \( \text{dur} \) is not present (i.e. \( \text{dur} = '\text{void}' \)), the attribute \( \text{begin} \) is always present (possibly equal to zero) and the attribute \( \text{end} \) is \( \text{void}, \text{indefinite} \) or a real number. The \( \text{PAR+BEGIN+END} \) rule described in Table 6 defines the semantics of the parallel composition in these cases. In the postcondition we make the components \( B_1 \ldots B_n \) evident to make it explicit that the postcondition should contain information about each \( c_i \). We can be a media object or a synchronization structure.

To prove the correctness of the tag \( <\text{par}> c_1 \ldots c_n </\text{par}> \), each \( c_i \) must be proven to be correct by assuming as its current time instant the current time instant of the parallel tag plus the offset given by the attribute \( \text{begin} \), i.e., if \( t_{cr}(c_i) = \text{init} \) is contained into the precondition of the tag \( c_i \), the precondition of each \( c_i \) must contain \( (t_{cr}(c_i) = \text{init} + k1) \) where \( k1 \geq 0 \) is the value of the attribute \( \text{begin} \) and \( \text{init} \) is the time instant at which the statement \( \text{par} \) is evaluated.

The evaluation of the end time instant of a par tag is a little more complicated, and not always possible. As a general remark, it is not possible to calculate the end time of a media item in two cases: if it is a static object and it does not have an attribute \( \text{end} \) or \( \text{dur} \) defined, or if it has an attribute \( \text{end} \) or \( \text{dur} \) equal to ‘\( \text{indefinite} \)’. In the same way, the ending time of a par statement cannot be calculated if its attribute \( \text{end} \) or \( \text{dur} \) is equal to ‘\( \text{indefinite} \)’, or it is not defined and one of its children has the attribute \( \text{end} \) (or \( \text{dur} \)) equal to ‘\( \text{indefinite} \)’.

Once we are able to decide whether a parallel composition terminates, we must calculate the time instant \( \text{stop} \). The semantics which describes the evaluation of \( \text{stop} \) is complex since different cases have to be considered. We discuss here what happens when an \( \text{end} \) attribute is defined: the case relating the \( \text{dur} \) attribute is very similar and will be discussed in the following.

We have to study four possible situations:

1. the tag \( c \) does not contain the definition of attribute \( \text{end} \) (i.e. \( \text{end} = '\text{void}' \)); in this case, the statement \( c \) ends when all its children (which are not static objects in \( \text{NotDur} \)) have finished their playbacks, i.e. at time instant \( \text{stop} = \max_{c_i} (\text{end}_B(c_i)) \);

2. all statements contained in the \( \text{par} \) tag end up before the \( \text{par} \) statement’s end, more precisely before time instant \( \text{init} + k2 \);

3. some continuous media items defined inside \( c \) have a natural duration wider than the duration of \( c \);

4. some items defined inside \( c \) have a duration, defined with an attribute \( \text{dur} \) or \( \text{end} \), wider than the duration of \( c \).

Cases 1, 2 and 3 are all correct. In the first two cases, each media object or statement within \( c \) lasts for a period of time equal or shorter than the duration of \( c \). If a static media item has not a duration defined, its duration is equal to the duration of \( c \). In case 3, if a continuous media \( c_i \) has a natural duration longer than the duration of \( c \), its playback will be truncated at \( c_i \)’s end.

Case 4 is not correct since the author gives a double, and contradictory, definition of the duration of media items involved, thus generating a temporal conflict. Note that case 4 includes also the case in which the parallel composition has a finite duration, but contains some children with an
indefinite duration, which is, by definition, longer than any other finite value.
We can apply the PAR+BEGIN+END rule in cases 1, 2 and 3, since the applicability conditions are satisfied. In case 1, all media items end before time instant stop because it is chosen as the maximum value. In case 2, all media items end before init + k2 = stop from the hypothesis. In case 3 all media items ending after the time instant stop belong to NotDur, therefore finite(k2) ⇒ ∀ci endci(c) ≤ stop ∨ cI endI(c) ≤ NotDur, hence the applicability condition is satisfied and the rule can be applied. The same applicability condition prevents us to apply the PAR+BEGIN+END rule in case 4 when a statement cI has a finite duration longer than c.

The statement finite(k2) ⇒ ¬Indef(c) states that in presence of a finite value of k2, the rule can be applied to the statement c only if it ends, i.e., it does not contain, at any level of nesting, an item with an indefinite duration.

Otherwise, the applicability condition Indef(c) ⇒ (¬defined(k2) ∨ finite(k2)) states that if the statement does not end and the attribute end is defined, then it must be equal to "indefinite". Finally the condition ∀ci begini(c) ≥ init+k1 expresses the fact that all children of c must start together with c or after it.

Let us illustrate how our rules find out temporal conflicts like the one described in case 4, due to an author's error which can happen when the structure becomes more complex, including a lot of tags nested one into the other. Let us consider the following tag:

```
<par id="p" begin="0" end="5s">
  <img id="i" begin="0" end="5s" />
  <text id="tx" begin="0" end="7s" />
</par>
```

Even if the temporal conflict is evident since the tag is simple, (text part tx lasts more then the tag in which it is contained), we try to check the semantic correctness of this statement to show how the system works.

We would like to prove that

\[ \{t_{cr}(p) = 0\} \text{par } p \ldots > (Q) \]

where \( Q \equiv \{ \text{begin}(i) = 0, \text{end}(i) \leq 5, \text{begin}(tx) = 0, \text{end}(tx) \leq 5, \text{begin}(p) = 0, \text{end}(p) = 5 \} \) but statement \( p \) is not correct since rule PAR+BEGIN+END (see Table 6) cannot be applied. In fact, since both tx and i do not belong to the set NotDur, in order to apply the rule we would have to prove the premises:

\[ S_i \equiv \{t_{cr}(i) = 0\} \ i \ \{\text{begin}(i) = 0, \text{end}(i) = 5\} \]
\[ S_{tx} \equiv \{t_{cr}(tx) = 0\} \ tx \ \{\text{begin}(tx) = 0, \text{end}(tx) = 5\} \]

The first triple \( S_i \) is valid and we can prove it by the axiom MEDIA+BEGIN+END+DUR, but we cannot prove the triple \( S_{tx} \) which is not valid. Therefore the PAR+BEGIN+END rule cannot be applied since the premise \( S_{tx} \) cannot be verified. In this case, the answer of our tool is that the presentation contains a semantic conflict since media item tx ends at time instant 7 while its father ends at time instant 5.

The rule which describes the semantics of the sequential composition is very similar to the PAR+BEGIN+END rule since the two tags can express the same synchronization if the values of the attributes are properly defined. It is presented in Table 7. Also in this case, we consider the attribute dur equal to void.

### Table 7: Proof rule for the sequential composition when the attribute dur is equal to void

With respect to the parallel composition there are only two differences: first, the current time instant of each child is equal to the end time instant of the previous child, and not to the current time instant of the seq tag. Second, the seq statement imposes a duration equal to zero to static media items which have not a defined duration, i.e., beginB(ci) = h and endB(ci) = h if ci is a static media contained in NotDur. This means that they are never played in the user screen.

So far we consider only the use of the attribute end, but, as already discussed for media item definition, statements can also contain an attribute dur whose semantics is very similar to the end attribute and therefore, an easily translation can be obtained with the rule CMD+BEGIN+END+DUR illustrated in Table 8.

### Table 8: Proof rule for a general composition of tags when the attribute dur is defined
3.2 Particular values for begin and end

The discussion so far consider only a time value (e.g. a number of seconds) as possible value for the attribute begin and end. SMIL language permits also other ways to define the starting or the ending time of tags using events (see Table 1). Let us consider the case in which the start (or the end) of a media, or a group of media items, occurs when the user keys in a character, say ‘s’, in the keyboard as described by the following tag:

\[<cmd c begin='accesskey(s)+k'/>\]

where \( accesskey(s) \) means that the user has to key in the character ‘s’ and \( k \geq 0 \) represents a number of seconds.

The correctness of this statement can be proven only if we already know the instant in which the event \( accesskey(s) \) takes place. Therefore, to define a rule for this kind of statements, we need a function \( t : A \rightarrow \mathbb{N} \) which takes as input an event \( accesskey(s) \in A \) and returns the time instant in which this event takes place; this function can be used in the preconditions of the statement to constraint the input event.

BEGIN+ACCESSKEY

\[ \{ A \} \left< cmd c begin='keyin+k'\ right> 'value'/\left\{ B \right\} \]

\{ A' \} \left< cmd c begin='acsk(s)+k'\ right> 'value'/\left\{ B \right\}

where \( A' = A \cup \{ t(acsk(s)) = keyin \} \)

APPLICABILITY CONDITION:

\( A' \implies \{ t(acsk(s)) \geq t_{\omega}(c) \} \)

END+ACCESSKEY

\[ \{ A \} \left< cmd c begin='value'ight> 'keyin+k'\left\{ B \right\} \]

\{ A' \} \left< cmd c begin='value'ight> 'acsk(s)+k'\left\{ B \right\}

where \( A' = A \cup \{ t(acsk(s)) = keyin \} \)

APPLICABILITY CONDITION:

\( A' \cup B \implies \{ t(acsk(s)) \geq begin(c) \} \)

Table 9: Proof rules for SMIL statements with an accesskey in the definition of the begin or the end attribute

Table 9 shows the rules to deal with statements with a begin or an end attribute which is bound to an input from the keyboard. Due to space constraints, \( acsk(s) \) is used instead of \( accesskey(s) \). Moreover, we write begin='value' or end='value' where value can assume any of the admitted values listed in Table 1.

These rules simply state that the input from the keyboard must occur after the evaluation of the statement, represented by \( t_{\omega}(c) \), or after its beginning if \( accesskey(s) \) is defined in the end attribute of a statement for which also a begin attribute is defined. Once this hypothesis is verified, since \( t(acsk(s)) = keyin \in A' \), the triple

\( \{ A' \} \left< cmd c end='accesskey(s)+k'\ right> \left\{ B \right\} \)

holds whenever we can prove that

\( \{ A \} \left< cmd c end='keyin+k'\ right> \left\{ B \right\} \).

Note that almost all other events could be addressed in the same way as soon as we assume the existence of a suitable function recording the time instant in which the event occurs. As an example, the activateEvent represents the time instant in which an user clicks on a media items, and therefore, from our point of view, it is not different from the user clicking on the keyboard.

Another possibility offered by the SMIL standard is to bind the begin (or end) event of a (group of) media item with the begin (or end) event of another (group of) media item \( n \). As an example, consider the tags

\[<par id='p'> <audio id='au' end='stop'/></par>\]

\[<par id='m'> <audio id='au' begin='n.begin+5'/></par>\]

in case (1), the whole par statement ends when media item \( au \) ends; in case (2) media item \( m \) begins 5 seconds after the beginning of \( n \).

Tag (1) can be considered similarly to the accesskey case: if we already know (from the premise) the end point of \( au \), i.e. \( end(au) = stop \), we can then analyze the tag \( <par id='p'> p=stop</par> \).

Therefore the following rule can be applied to tag (1):\n
\[\{ A \} \left< par c end='stop+k'\right> c_1..c_n</par> \left\{ B \right\}\]

\[ \{ A \} \left< par c end='ci.end+k'\right> c_1..c_n</par> \left\{ B \right\} \]

APPLICABILITY CONDITION:

\( end_{\omega}(c_i) = stop \)

As we have already said, situation is more complex in case (2), which cannot be analyzed singularly since its evaluation needs information about the begin of media item \( n \). For this reason we must consider a set of media items as shown by the following rule:

BEGIN+EVENT

\[ \{ A \} \left< cmd c\right> c_1..c_n</cmd> \left\{ B \right\}\]

\[ \{ A \} \left< cmd c\right> c_1..c_n</cmd> \left\{ B \right\} \]

where \( n \in Closure(c_i) \), \( d \equiv \{ cmd id='m' begin='n.begin+k'\}/ \in Closure(c_j) \), and \( c_j \) is obtained from \( c_j \) by replacing \( d \) with \( d' \equiv \{ cmd m begin='begin_{\omega}(n)+k' \}/ \).

In this paper we consider a limited set of combinations of attributes and events, but the approach is easily generalizable to cover all the possibilities offered by the standard.

3.3 The excl tag

SMIL language provides also a tag for the exclusive composition of media items, i.e., the tag excl, whose semantics states that only one of its children is active at any given time instant. Therefore, this tag is very similar to the sequential composition, even in this case only one child is active at a time, but excl does not impose any order in the visualization of the children. This means that each child may contain the attribute begin in the definition, or may be activated by the user, e.g. following a link. Let us consider the following example:

\[<par><img id='a'/><img id='b'/><img id='c'/></par>\]

\[<excl id='e'>dur='10'</excl>\]

\[<video id='video_a' begin='a.activateEvent'/><video id='video_b' begin='b.activateEvent'/><video id='video_c' begin='c.activateEvent'/></excl>\]

in this case, the user chooses a video clip by clicking on an image button chosen between media items a, b and c.
The corresponding video is activated by the proper activateEvent. The excl tag simply states that only one video clip plays at a time: in fact, the video currently playing is stopped when the user clicks on another image, choosing another video clip. The example shows how the excl command does not deal with the activation of its children but with its deactivation; in fact, the playback order of the video clips completely depends on the user choices and not on the tags’ definitions.

The semantics of the excl tag is described in Table 10 by rule EXCL+BEGIN+END. Like tags par and seq, the excl tag begins at its current time instant, or after k1 time instants if the attribute begin is finite, and ends, when there are no children playing. This means that, it can have an instantaneous duration if no child starts together with it. For this reason, the attribute end of this statement usually does not contain the special value ‘’void’’.

The EXCL+BEGIN+END rule is very similar to the rule which describes the semantics of parallel composition, therefore we do not repeat here the problem of the termination of the tag. Even in this case, to prove the correctness of the statement ⟨excl⟩ c1 . . . cn ⟨/excl⟩, each ci must be proven to be correct, assuming as its current time instant the current time instant of its father. Since the exclusive tag may impose a premature stop of the playback of its children, in some cases, we do not require to know the time instant in which the child ci ends in the premises, i. e.:

1. when ci ends together with excl (i.e. k2 is finite) if it does not contain the attribute end or dur in its definition (i.e. ci ∈ NotDur);
2. when the playback of ci is stopped before its natural termination due to the user interaction or some other external event (i. e. ∃j [beginB(cj) = ti]).

The applicability condition prevents the application of the rule in presence of temporal conflicts. Among the conditions already discussed for the parallel composition, the condition ∀ci, cj (beginB(ci) ≤ beginB(cj)) ⇒ (endB(ci) ≤ beginB(cj)) states that only one child plays at any given time instant, i. e., if child ci begins before child cj, it also ends before cj’s beginning.

## 4. EQUIVALENCE OF SMIL TAGS

In this section we introduce two notions of equivalence between SMIL tags: an observable equivalence, ≈, and a strong equivalence, ≡S. The first one formalizes the informal idea that two tags are equivalent if a user cannot distinguish them just by observing the playback of the various media items in her/his screen. The second one is stronger because it specifies when two tags can be substituted each other. For this reason it also asks the same reference for the two tags, i. e., the same value for the function tcr.

First we introduce an equivalence on assertions. We say that two assertion P1 and P2 are player-equivalent, and write P1 ≈P P2, if they contain the same constraints on media items.

**Definition.** Let P1 and P2 be assertions. P1 ≈P P2 if for every function f(·) ∈ {dur, begin, end} and media m:

\[(f(m) = k) \in P_1 \leftrightarrow (f(m) = k) \in P_2.\]

Then we can introduce the notion of observable equivalence between tags.

**EXCL+BEGIN+END**

\[\{A \cup \{tcr(c_i) = init\_c_i\}\} \cup \{B_i\} \forall 1 \leq i \leq n\]

\[\{A’\} \langle excl \ c \ attribute-list> c_1 \ldots c_n \langle/ excl\rangle \{B\}\]

where

attribute-list = begin='k1' end='k2'dur='void'

\[A’ = \bigcup_{i=1}^n A_i \cup \{tcr(c_i) = init\}
\]

\[\begin{align*}
\text{init}_{c_i} &= \text{init} + k1 \\
B &= \bigcup_{i=1}^n B_i \cup \{\text{begin}(c_i) = \text{init} + k1\} \cup \text{End}
\end{align*}
\]

\[\begin{align*}
\text{stop} &= \left\{ \begin{array}{ll}
\max_{c_i}(\text{endB}(c_i)) & \text{if \ } \text{finite}(k2) \\
\emptyset & \text{otherwise}
\end{array} \right.
\]

\[\text{End} = \left\{ \begin{array}{ll}
\{\text{end}(c) = \text{stop} & \text{if \ } \neg \text{Indef}(c)
\end{array} \right.
\]

\[\begin{align*}
B_i’ &= \left\{ \begin{array}{ll}
B_i \setminus \{\text{end}(c) = t_i\} & \text{if } \exists j \text{ beginB}(c_j) = t_i \\
\forall (\text{finite}(k2) & \wedge c_i \in \text{NotDur}) \\
\text{otherwise}
\end{array} \right.
\]

**APPLICABILITY CONDITION:**

\[\text{finite}(k2) \implies \neg \text{Indef}(c)
\]

\[\neg \text{Indef}(c) \implies (\neg \text{defined}(k2) \lor \text{indefinite}(k2))
\]

\[\text{finite}(k2) \implies \forall c_i, (\text{endB}(c_i) \leq \text{stop} \lor c_i \in \text{NotDur})
\]

\[\forall c_i, \text{beginB}(c_i) \geq \text{init} + k1
\]

\[\forall c_i, c_j (\text{beginB}(c_i) \leq \text{beginB}(c_j))
\]

\[\implies (\text{endB}(c_i) \leq \text{beginB}(c_j))
\]

**Table 10: Proof rule for the exclusive composition when the attribute dur is equal to void**

**Definition (Observable Equivalence).** Let c1 and c2 be SMIL tags. Then c1 ≈ c2 if:

- for any pair P1, Q1 such that ⊨ {P1}c1{Q1} there exist P2, Q2 such that P1 ≈m P2, Q1 ≈m Q2 and ⊨ {P2}c2{Q2};
- for any pair P2, Q2 such that ⊨ {P2}c2{Q2} there exist P1, Q1 such that P1 ≈m P2, Q1 ≈m Q2 and ⊨ {P1}c1{Q1}.

As an example, consider the following commands:

\[c_1 \equiv \langle img \ id="im2" \ begin="10" \ dur="5s" /\rangle
\]

\[c_2 \equiv \langle img \ id="im2" \ dur="5s" /\rangle.
\]

We can prove \(\{P_1\}c_1\{Q\}\) only if \(P_1 \equiv A \cup \{t_{dur}(im2)= start – 10\} \) and \(Q_1 \equiv A \cup \{\text{begin}(im2)= start, end(\text{im2})= start + 5\}\), for some assertion A; similarly, we can prove \(\{P_2\}c_2\{Q\}\) only if \(P_2 \equiv A \cup \{t_{dur}(im2)= start\} \), \(Q_2 \equiv A \cup \{\text{begin}(im2)= start, end(\text{im2})= start + 5\}\), for some assertion A. Hence c1 and c2 are observable equivalent.

The notion of observable equivalence correctly captures the behavior of the overall execution of a SMIL tag but it is not sufficiently strong to induce a substitution property. Consider for instance the following tags:

\[d_1 = \langle seq \ id="s" \rangle
\]

\[\langle img \ id="im1" \ dur="7s" /\rangle
\]

\[\langle img \ id="im2" \ dur="5s" /\rangle
\]

\[\langle seq\rangle\]

\[d_2 = \langle seq \ id="s" \rangle
\]

\[\langle img \ id="im1" \ dur="7s" /\rangle
\]

\[\langle img \ id="im2" \ begin="10s" \ dur="5s" /\rangle
\]

\[\langle seq\rangle\]
They are clearly not equivalent even if they differ only for the replacement of the tag \(c_1\) with the observable equivalent tag \(c_2\).

The problem depends on the fact that the notion of observable equivalence does not impose a fixed “starting” time instant for a SMIL tag, that depends on the time in which it is analyzed. The function \(t_{\text{start}}\) sets the “starting” time of a SMIL tag when it is nested into another tag. Therefore a tag can be replaced with another one, only if their \(t_{\text{start}}\)-values are equal.

Hence we need a stronger notion of equivalence if we want to prove a substitution property.

**Definition (Strong Equivalence).** Two tags \(c_1\) and \(c_2\) are strong equivalent, \(c_1 \approx_S c_2\), if they are observable equivalent, \(c_1 \approx c_2\), and they are “time coherent”:

\(t_{\text{start}}(c_1) = \text{init} \) \(\in\) \(P_1\) \(\iff\) \(t_{\text{start}}(c_2) = \text{init} \) \(\in\) \(P_2\)

for all \(t \in \{\text{begin}, \text{end}\}\) \(f(c_1) = k \) \(\in\) \(P_1\) \(\iff\) \(f(c_2) = k \) \(\in\) \(P_2\).

Note that, in the definition of strong equivalence the condition, for all \(t \in \{\text{begin}, \text{end}\}\), \(f(c_1) = k \) \(\in\) \(P_1\) \(\iff\) \(f(c_2) = k \) \(\in\) \(P_2\) seems to be pointless since \(Q_1 \approx_m Q_2\). This is true when tags \(c_1\) and \(c_2\) start and end with a media item.

This is not true when there are time intervals in which no media items are played at the beginning or ending of the tag, or when \(\sim\text{Indef}(c_1) \wedge \sim\text{Indef}(c_2)\). Consider as an example the tag:

\[
\begin{array}{c}
\text{<seq id="s" begin="10s"/>}
\end{array}
\]

where no media item plays for the first ten seconds. In this case, the condition \(\{\text{begin}(c_1) = \text{bc}, \text{end}(c_1) = \text{ec}\} \subseteq Q_1 \iff \{\text{begin}(c_2) = \text{bc}, \text{end}(c_2) = \text{ec}\} \subseteq Q_2\) must hold to find out a strong equivalent tag.

\[
\begin{array}{c}
\text{Note here that all the rules of the proof system can be used both for a top-down construction of a correct playback sequence of the media items involved in the multimedia presentation and for a bottom-up analysis of the SMIL document. This second feature is particularly useful during the context adaption process; it minimizes the set of preconditions needed to evaluate a SMIL tag, i.e., the natural duration of the continuous objects, if present, and}
\end{array}
\]

\[
\begin{array}{c}
\text{the compositionality of the approach allows for an easy extension of SMIL features actually considered.}
\end{array}
\]

5. **CONCLUSIONS AND RELATED WORK**

This paper presents a formal semantics for the temporal aspects of SMIL documents. We start by defining a set of inference rules inspired by the Hoare triples which describe how the execution of a piece of code changes the state of the computation. A prototype implementation of a computer-aided authoring system, including the Semantic Validator Module based on the proposed semantics, is currently under development.

This paper mainly focuses on SMIL 2.1 features but since SMIL 3.0 specification leaves the basic syntax and semantics of the SMIL 2.1 timing model unchanged [3], it also applies to the latest version.

The main advantages of this work are the following:

- It allows the author to check the consistency of a multimedia presentation based on the SMIL standard and not on a proprietary format;
- It assists multimedia authoring by pointing out conflicting values in the document;
- It allows for a modular evaluation of the tags nested in a SMIL document and helps the context adaption process;
- It minimizes the set of preconditions needed to evaluate a SMIL tag, i.e., the natural duration of the continuous objects, if present, and
- The compositionality of the approach allows for an easy extension of SMIL features actually considered.

Module: It is sufficient
to translate the tag to an equivalent one which chooses the minimum value for the attribute. This also means that our approach is open to future modifications of SMIL.

Other works in literature study a way to find out temporal conflicts into SMIL documents. In [17, 16, 19], Sampaio et al describe RT-LOTOS, a formal description of SMIL tags which enables the generation of a valid scheduling for its rendering, considering QoS problem. The authors do not aim at defining a semantics for SMIL language, but compare different players’ behaviors which are still implementation-dependent.

Yang [20] and Yu [21] proposes the use of Petri Nets to describe the temporal evolution of a SMIL document. Yang translates the SMIL synchronization tags into transitions and places of the Real Time Synchronization Model (RTSM) and tries to detect possible temporal conflict, but this work is limited to the features of SMIL 1.0. Yu defines a formalism based on Petri Nets named SAM (Software Architecture Model) which aims to check if QoS properties, expressed through logical formulas, are satisfied, and not to the verification of the semantic correctness of the SMIL document.

The only real attempt to define a formal semantics for SMIL is presented in [10] by Jourdan. This approach is based on the use of timed automata and has been used during the design of SMIL 2.0 to improve specification, since the author was a co-editor of the document which describes timing and synchronization features of this language. The work presented in this paper mainly focuses on SMIL 1.0 and take into account only two new features of SMIL 2.0.

Other works address the problem of temporal consistency of multimedia documents not described with SMIL language. Among others, Elias [6] presents an algorithm, based on the graph theory, which is able to dynamically maintain a consistent and complete set of constraints during the authoring phase. Other works address the same problem with constraints solver techniques. Differently from our approach, all the works addressed here require to translate the SMIL document into another formalism, e.g., a set of temporal constraints or a Petri Net, in order to check its temporal consistency. This operation is not always cost-effective, especially when the complexity of the input file increases and a non compositional approach is used.

Finally, since most available authoring systems adopt “... SMIL language for building the final representation scheme” [4], we argue that a formal semantics for this language is needed.

6. ACKNOWLEDGMENTS

The authors would like to thank Maria Luisa Sapino and the anonymous referees for their helpful suggestions.

7. REFERENCES